DEC STD 073-3 Electronic Media Master for Software Engineering

DOCUMENT IDENTIFIER: A-DS-EL00073-03-0000 Rev A, 31-Jan-1992

ABSTRACT: This section of DEC STD 073 defines the format and structure requirements of files that serve as the masters for electronic media production at Digital. Two data formats are specified: one for disks and another for tapes. The formats specified are not dependent on the methodology and format used in the original creation of physical disk or tape masters.

APPLICABILITY: The requirements of this standard are mandatory for software developers, and others who are responsible for producing electronic media masters. The responsibilities detailed in this document are applicable only to those groups developing and manufacturing 32-bit product sets.

STATUS: APPROVED 31-Jan-1992; use VTX SMC for current status.

This document is confidential and proprietary, and is the property of Digital Equipment Corporation. It is an unpublished work protected under applicable copyright laws.

© Digital Equipment Corporation. 1992. All rights reserved.

DEC STD 073-3 Electronic Media Master for Software Engineering

DOCUMENT IDENTIFIER: A-DS-EL00073-03-0000 Rev A, 31-Jan-1992

REVISION HISTORY:	No previous revision exists.
Document Management Category:	Cross Documentation Vocabulary and Practices (TDV)
Responsible Department:	U.S. Software Supply Business
Responsible Person:	Ted Luszey

APPROVAL: This document has been reviewed and recommended for approval by the General Review group for its category.

Ted Luszey - Electronic Publishing Engineering Business

Eric Williams - Standards Process Manager

Direct requests for further information to:

Ted Luszey
Use $ VTX ELF for the latest location information.

Use VTX SMC to order copies of this document from Standards and Methods Control. Send distribution questions to JOKUR::SMC or call DTN: 234-4423.

The DIGITAL logo is a trademark of Digital Equipment Corporation.

CONTENTS

Preface v

Chapter 1 INTRODUCTION 1–1

1.1 Purpose 1–1

1.2 Scope 1–2

1.3 Responsibilities 1–2
1.3.1 Engineering 1–2
1.3.2 Manufacturing 1–3

Chapter 2 SERIAL MEDIA MASTER FILE 2–1

2.1 Magtape Media Master File Design Rationale 2–1

2.2 Structure of the Magtape Media Master File 2–1

Chapter 3 RANDOM ACCESS MEDIA MASTER FILE 3–1

3.1 Random Access (Disk) Media Master File Design Rationale 3–1

3.2 Data Compression in Random Access Media Master Files 3–1

3.3 Structure of the Random Access Media Master File 3–2

Appendix A HEADER BLOCK FORMAT A–1

Appendix B OBJECT DESCRIPTOR BLOCK B–1

B.1 Format of the Object Descriptor Block B–1

B.2 Format of the Tape Object Descriptors B–2

B.3 Format of the Disk Object Descriptors B–4

Appendix C REFERENCED DOCUMENTS C–1

C.1 EL-Class Digital Documents C–1

FIGURES

1 Example Tape ODB Chain 2–2
2 Example Disk ODB Chain 3–3
3 Header Block Format A–2
4 Object Descriptor Block Format B–1
5 Tape Object Descriptor Formats B–3
6 Disk Object Descriptor Formats B–4

Preface

DEC STD 073 Manufacturing and Packaging for Publications establishes the specifications for publications and film products. While other sections of *DEC STD 073* define physical masters and the resulting hard copy documents, this section defines electronic media masters.

This standard defines the structure and production information content of files that serve as the masters for all media production at Digital. Two data formats (one for disks and another for tapes) that are not dependent on the methodology and formats used in the creation of the physical disk or tape masters are specified.

NOTE

Readers of this document should be familiar with the jargon and technologies of the media production process.

NOTE

All numeric values throughout this document, unless specified otherwise, are decimal (base 10) values. This includes byte offset locations within data structures as well.

CHAPTER 1

INTRODUCTION

The text in this chapter provides detailed information regarding the goals and content of this standard and the responsibilities of the internal groups that produce electronic media masters.

1.1 Purpose

This standard specifies independent formats that support the unique requirements of software publishing. Optimization for high speed replication is a unique requirement of software publishing and not normally a key consideration for software developers. The formats specified in this standard are highly optimized to support the required high speed replication processes.

Specifically, this document provides the requirements for the following.

- Structure of media master files (one format for disk and another for tape)
- Information content of the files

The specific goals of this standard are the following.

- Define an electronic media master format that can be used for all serial magnetic media, for example, 1/2-inch tape, TK50, and others
- Define a second electronic media master format that can be used for all random access media, for example, RK07, RA60, TU58 tape, floppies, and others
- Define a standard that will enable and support simple, highly efficient, production site specific (if desired) Copy/Verify routines
- Define a standard that will enable and support distributed media mastering
- Define a standard that will enable and support direct master production (that is, without creating a physical media master that is subsequently mastered)
- Define electronic media master formats that are not dependent on the methodology and format used in the original creation of physical disk or tape masters
- Define standards and processes that are robust and predictable enough to allow for the safe elimination of engineering media verification
- Define electronic media master formats that are complete in terms of the information necessary to accomplish the preceding goals

Note that the format of data files for intake into the Consolidated Distribution process is not specified by this standard.

1.2 Scope

This standard comprises one component of the architecture for software publishing at Digital. Within this document, the term *software* refers to one of the following.

- The product, which includes documentation as well as the executable code (binaries)
- The executable code only

The intended meaning of the term will be obvious from the context.

Within this standard, the term *media* refers to disks, floppies, and magnetic tapes.

Revision of this standard is directed toward Digital internal processes. However, eventually, the internal processes may be applied to the external software supply market. Future revisions of this document can address that external market more explicitly as required.

Copy-protected media (typically floppies) use techniques such as sector scrambling to protect the information on the media from being copied. This standard does not define data structures that carry sector-specific information. Being able to produce and reproduce such media requires special hardware. Digital has not invested in such hardware, preferring to stock the finished goods supplied from the original vendor. Digital does not copy protect any media.

This specification could be upgraded to provide the appropriate level of support to cover copy-protected media. Such an upgrade would result in significant modifications to the random access media section of the standard.

1.3 Responsibilities

Text under subheads 1.3.1 and 1.3.2 details the responsibilities of internal groups involved in the media master file production process. Note that these responsibilities are applicable only to those groups developing and manufacturing 32-bit product sets. The process and responsibilities applicable to other than 32-bit product sets will be defined in a future revision of this standard.

1.3.1 Engineering

- Engineering shall prepare and deliver the save sets, now called Self-Contained Units (SCUs), and the control information necessary for manufacturing.
- Engineering shall be responsible for the quality and completeness of media masters, both physical and electronic, that are delivered to Manufacturing.

1.3.2 Manufacturing

- Manufacturing shall transform each SCU into one or more media master files, compliant with this standard, by executing the media creation application(s).
- Manufacturing shall be responsible for exactly duplicating the contents of the media master file(s) onto the appropriate physical media.
- Manufacturing shall be responsible for producing the initial Bill of Materials (BOM).

CHAPTER 2

SERIAL MEDIA MASTER FILE

This chapter provides the design rationale and describes the structure of the serial (magtape) media master file. Figure 1 shows the data block chain that describes the structure of the input tape.

2.1 Magtape Media Master File Design Rationale

The primary goal of the magtape media master format is to capture both the context and content information of the tape. Context information describes the tape, for example, tape length, density, and type. Content information describes the records, for example, data record or end-of-file (EOF) mark order and record lengths. Content information also includes the data. From the context and content information in the media master file, a Copy/Verify process must be able to produce an exact replica of the original tape.

The media master format must also meet the following stringent performance goals.

- It must be quickly readable to enable driving high performance Copy/Verify processes, such as multiple high-speed streaming tape processes.
- It must support the direct media mastering process, which requires that the data structure be efficiently and bidirectionally traversable.
- Format of the data structure must be compact and require little working storage to manipulate it.

2.2 Structure of the Magtape Media Master File

Each magtape media master is represented by one disk file. The file shall be structured to consist of the following elements.

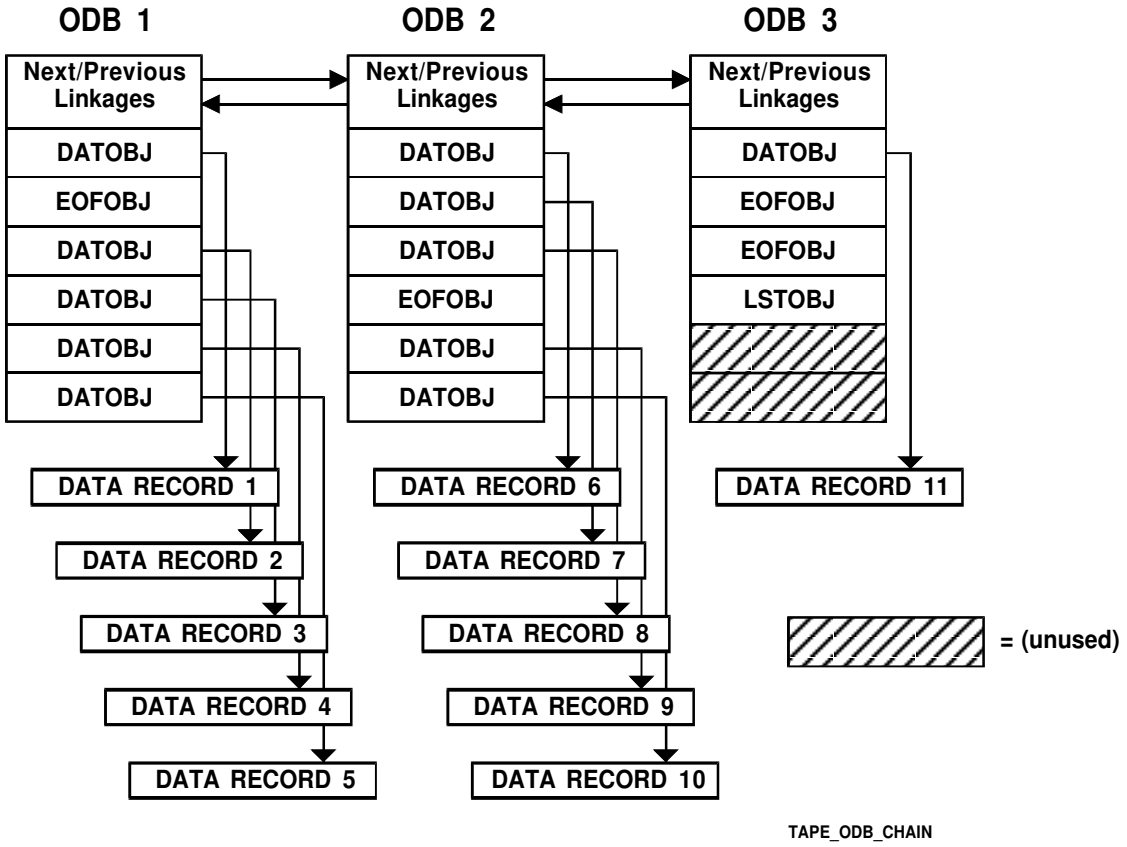
- Header block contains information about the tape such as the master file format, revision level, mastering information, part number, size of the largest record contained on the tape, and label text.
- Object descriptor blocks (ODB) describe the structure of the tape by recording the sequence and size of the records, and the location of EOF marks.
- Data blocks contain the actual data read from the records on the tape.

The ODB is the master file data structure that describes how the data is organized on the tape. The structure of the ODB is designed to maximize the speed of Copy/Verify processes rather than the mastering process although the ODB is efficient in the mastering process also.

Each ODB shall be one disk block long (512 bytes) and form a linked list that threads through the master file. The ODB shall be filled with two longword entries called object descriptors. Object descriptors primarily describe data records and EOF marks. Each ODB shall contain slots for 63 object descriptors. The first two longwords in each ODB shall be reserved for the pointers to the next and previous ODBs in the list.

Figure 1 shows an ODB chain for a short magtape. Note that the ODBs in the figure have only 6 descriptor slots instead of the usual 63.

Figure 1: Example Tape ODB Chain



In Appendix B, Figure 4 shows the format of the ODB and Figure 5 shows the format of the object descriptors themselves. Also, Appendix B provides a full description of these data structures.

The first ODB shall be located at virtual block number 2 (VBN 2) in the file. Note that VBN 1 is reserved for the header block. The data for the first data record described in ODB 1 shall start at VBN 3 and continue for as many blocks as necessary to hold the data. For example, an 80-byte record would be entirely contained in one disk block. An 80-byte record would also consume one disk block. That is, the next data record following the 80-byte record would be stored starting at the next block in the file rather than at the 81st byte of the partially filled block. While some space is wasted in the master file as a result

of this technique, the data can be accessed much faster during the Copy/Verify process because data deblocking is not required.

Each data record written to the master file shall use the next available slot in the current ODB. Data records are described by data object descriptors. See Figure 5 for the format of the tape object descriptors. Each data object descriptor shall record the starting VBN of the data location in the master file as well as the length of the data in bytes.

EOF marks on tape are recorded in the ODB as EOF descriptors. EOFs consume only one ODB slot since there is no data associated with an EOF mark.

An ODB shall be written to the master file when all of the slots in an ODB are filled with descriptors or when the end of the tape being processed is reached. The ODB is written after the pointers in the ODB are updated to point to the next and previous ODB in the list. For the first ODB, the value in the previous ODB field is 0. For the last ODB, the value in the next ODB field is also 0.

The last descriptor in the last ODB of Figure 1 is a special descriptor that instructs the Copy/Verify program when there is no more data to process. If this descriptor were not used, Copy/Verify programs would have to contain the intelligence to recognize when the end-of-volume had been reached and to stop processing. Because mastering programs must contain this intelligence, there is no need for Copy/Verify programs to duplicate it. Furthermore, in order to maintain the format independence of electronic media masters, Copy/Verify programs *should not* have the necessary intelligence to recognize the end-of-volume on a tape. This function is appropriate only to the mastering program(s) which usually *are* dependent on the format and/or context of the incoming data.

Figure 3 shows the format of the header block. This data structure shall be written by the mastering program. The header block records mastering information, the part number, the type and length of tape, the recording density, labeling information, and so on. This information is obtained from several sources such as the operator, the tape drive, and the data on the tape. See Appendix A for a full description of the header block.

CHAPTER 3

RANDOM ACCESS MEDIA MASTER FILE

This chapter provides the design rationale and describes the structure of the random access (disk) media master file. Figure 2 shows the data block chain that describes the structure of the input disk.

3.1 Random Access (Disk) Media Master File Design Rationale

The design rationale for the random access (disk) media master file is much the same as for the serial media master file since the same goals must be met. The disk media master must capture the data in a format that meets the following performance goals.

- Independence of data organization
- Compact (requiring little working storage)
- High-speed replication support

Just as for magtape media master files, format independence for disks is achieved by simply making a bit-for-bit image copy of the input disk. Every block on the master disk, even those not in use, is reproduced on the output disk.

Note that an output disk that has bad blocks cannot be accepted. An output disk with bad blocks cannot be used because every block on the output disk is explicitly written by the Copy/Verify program, and the data cannot be repositioned on the output disk. Because of the expense, guaranteed error free media is not required. However, any output disk with bad blocks must be rejected and used for some other purpose.

3.2 Data Compression in Random Access Media Master Files

To meet the storage performance goal, the disk media master format uses a compression scheme to represent large repeated data patterns. Use of data compression applies to any process that creates the master file, whether the source is a physical disk being mastered or an SCU being transformed directly into master files.

The raw data to be output to the master file must be searched for repeating data patterns. The pattern must be represented in a longword, and the pattern must fill one or more contiguous disk blocks. Repeating data segments of only one block in length should be considered because even one-block segments can be compressed by a factor of 64:1.

Note that new compression patterns can be chosen at any time without affecting Copy/Verify programs. These programs are required to react dynamically to any pattern that appears in a compression object descriptor. Note also that pre-allocation of buffers with the most commonly seen patterns would greatly improve the performance of a Copy/Verify program. These patterns could be the common *bad block locator* patterns, for example.

It is important to stress that compressed blocks must still be written to the output disk. Detection of a series of blocks containing, for example, one of the bad block patterns does not necessarily mean that the blocks are not in use. They could be an initialized buffer in the task image of the bad block locator utility. *No assumptions can be made.* Every block on the output disk must be written.

A cost is associated with the reduction of the size of the media master file through data compression. The primary cost is in the complexity and performance of any program that creates master files. The speed of the program is affected by searching for more patterns, which increases processing time. This increase in processing time is justified however, because a disk is usually mastered only once; therefore, the cost is usually incurred only once.

The other performance-related cost is in the Copy/Verify program. The program must dynamically create and populate buffers with the pattern of blocks that are compressed out of the master file. This pattern should not affect performance significantly if the Copy/Verify programs are well constructed, for example, if buffers are pre-allocated with the most commonly encountered patterns.

3.3 Structure of the Random Access Media Master File

The format for random access (disk) electronic media masters is a slightly modified version of the format specified for serial devices (tapes). Like tapes, each disk is represented by one master file. The file shall be structured to consist of the following elements.

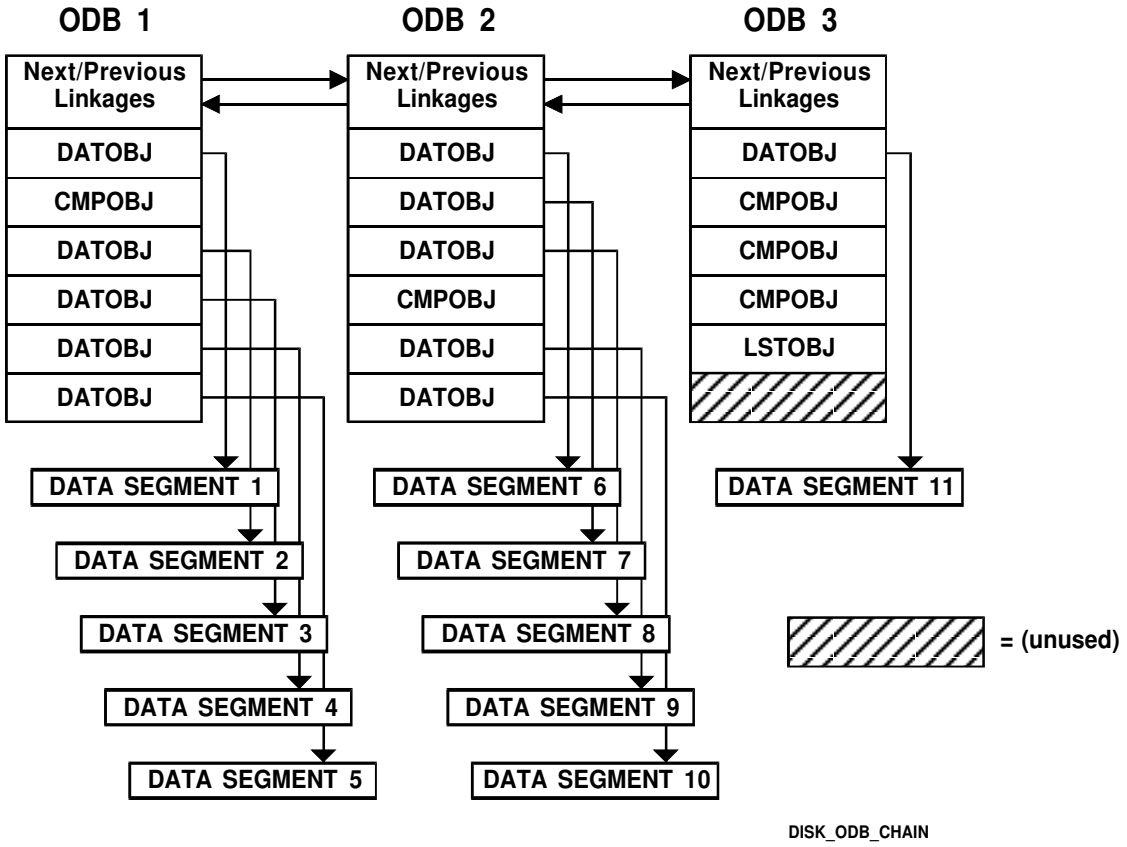
- Header block contains information about the disk, such as the master file format revision level, mastering information, part number, size of the disk, and label text.
- Object descriptor blocks (ODB) describe how the data is organized by recording the locations of data segments and compression segments.
- Data segments contain the actual uncompressed data read from the disk.

The master file data structure that describes how the data is organized on the disk is, the same as for tape, the ODB. The same basic ODB structure is used for both disk and tape; only the format of the object descriptors is slightly different. In fact, there is so little difference between the descriptor formats for disk and tape that one Copy/Verify program could handle both disks and tapes.

Each ODB shall be one disk block long (512 bytes) and form a linked list that threads through the master file. The ODB shall be filled with 2 longword entries called object descriptors. Object descriptors primarily describe data segments and compression segments. Each ODB shall contain slots for 63 object descriptors. The first two longwords in each ODB shall be reserved for the pointers to the next and previous ODBs in the list.

Figure 2 shows an ODB chain for a typical disk. Note that the *ODBs* in the figure have only 6 descriptor slots instead of the usual 63.

Figure 2: Example Disk ODB Chain



The first ODB shall be located at virtual block number 2 (VBN 2) in the file. VBN 1 shall be reserved for the header block. The data for the first segment record described in ODB (1) shall start at VBN 3 and continue for as many blocks as necessary to hold the data. For example, if the first data segment were 20 blocks long, it would be contained in virtual blocks 3 through 23 of the master file.

Each data record written to the master file shall use the next available slot in the current ODB. Data records are described by data object descriptors. Refer to Appendix B, Figure 6 for more details regarding the format of the data object descriptors. Each data object descriptor records the starting VBN of where the data was written in the master file, as well as the length of the data in bytes. For disks, this length will always be a multiple of 512 bytes.

Compressed data shall be recorded in the ODB by a compression object descriptor. Compression object descriptors consume only one ODB slot since the compression pattern is stored as part of the descriptor. The Copy/Verify program must use the compression pattern and the data segment length to build a buffer dynamically of the specified length and data pattern. This buffer is then output to the disk starting at the next logical block to be written.

Note that for both data and compression objects, the segment length field limits the size of the I/O buffer to a maximum of 128 blocks (512 bytes per block). This limit means that a data or compression segment that exceeds the 128-block limit must be represented by multiple objects.

The general management (filling, linking, terminating) of ODBs by the mastering program is the same for disks as it is for tapes. Please refer to the tape description in Chapter 2.

Figure 3 shows the format of the header block. This data structure shall be written by the mastering program. The header block records mastering information, the part number, the structural details of the disk, labeling information, and other information. This information is obtained from several sources such as the operator and the disk drive. See Appendix A for a full description of the header block.

APPENDIX A HEADER BLOCK FORMAT

Figure 3 shows the format of the header block. Information contained in the header block is obtained from several sources: the operator, the tape or disk drive, and from the data itself. Much of the header block is common for both the serial (magtape) and random access (disk) media master files. The sections that are unique to a particular media type are identified in the text that follows.

The header block shall contain the following fields with unused locations being zero filled.

- Master File Format Information field is a 2-byte field broken into the master file format value (byte 0) and format revision level (byte 1). For a magtape master file, the master file format field contains the binary value 1; for a disk master file, the binary value 2.

  The Revision Level field contains the binary value of the appropriate (magtape or disk) master file format. See the following table for the Revision Level values for this revision of the standard.

Standard Version	Disk Revision Level	Tape Revision Level	Changes from the Previous Version
Rev A	1	1	

- Mastering Program Information field contains the name and revision level of the program used to create the master file. The field is a counted ASCII string that starts at byte 2. Byte 2 contains the binary string length. The string itself starts at byte 3 and is limited to 19 bytes in length.

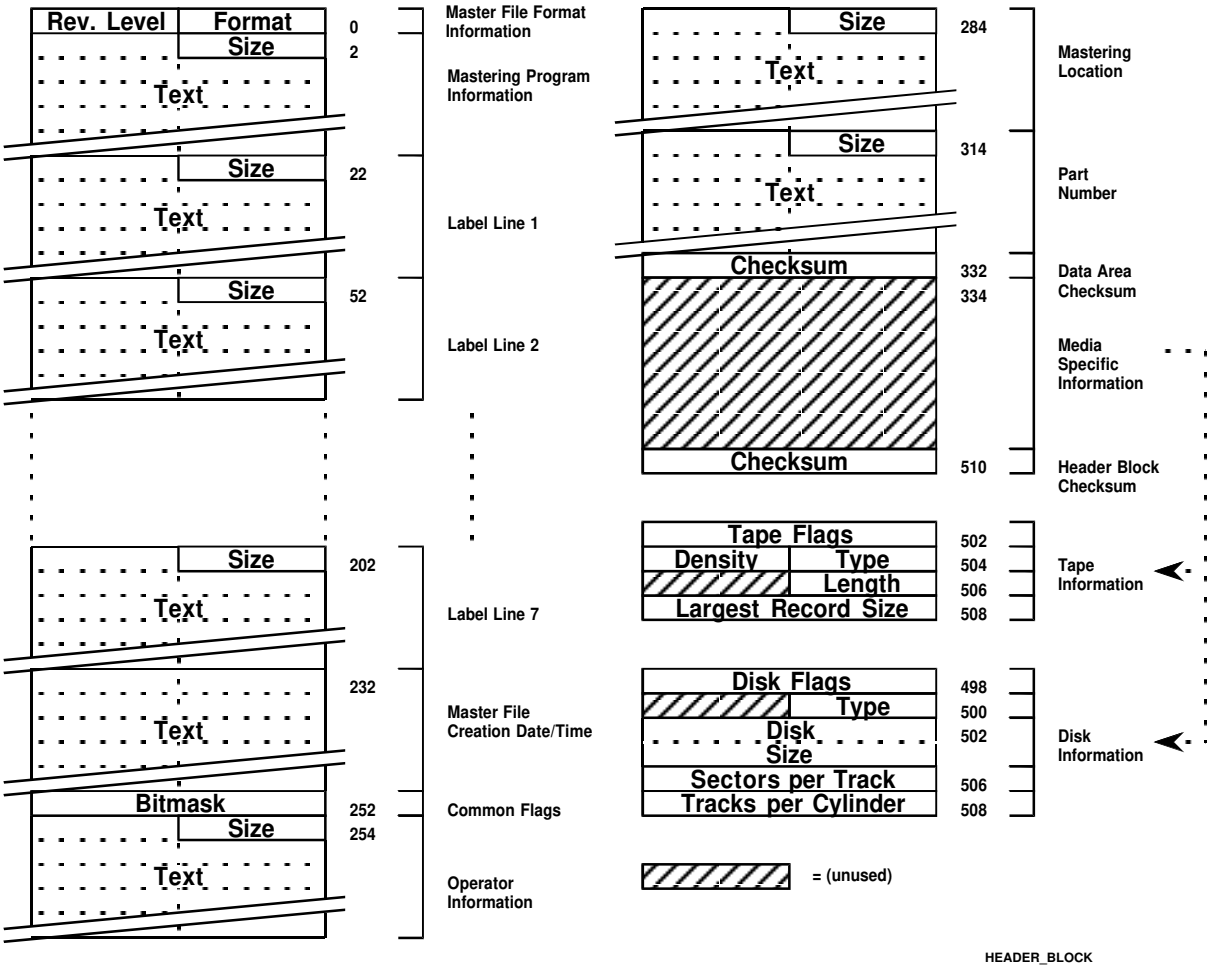
  NOTE

  Unused locations in a counted ASCII string are zero filled.

- Label Text field is a series of seven counted ASCII strings that contain the label text information. These fields start at bytes 22, 52, 82, 112, 142, 172, and 202. Each string may not exceed 29 bytes in length. All label printing programs shall look for 7 label lines and print any label line that is not null.

  The capability to store and use labels externally to the media master image is provided by a flag defined in the Common Flags field. The flag setting indicates whether the label text stored within the media master file should be ignored in favor of an externally stored label. This capability allows the same image to be used with labels written in different languages. For the same media master file, this flag may be set differently from one manufacturing location to another. Refer to the Common Flags field description and bit definitions table that appear later in this appendix.

Figure 3: Header Block Format



- Master File Creation Date and Time is an ASCII field that starts at byte 232 and is written in the following format: DD-MMM-YYYY␣HH:MM:SS (where ␣ is the space character).
- Common Flags is a binary word field that starts at byte 252. The flag word contains a bit mask of logical values that are common to all media master formats. The defined bits are shown in the following table.

Bit Number	Definition (0 = False, 1 = True)
0	Write protect the media
1	Serialize the label
2	Serialize the media
3	Produce a diagnostic label
4	External label(s) may be used

  Unused bits are reserved for future use.

  Bits 1 and 2 shall indicate whether to serialize the label and media, respectively. The actual serial numbers shall be stored and managed locally. That is, each manufacturing site shall maintain a data base that relates component 2-5-2 numbers to media serial numbers for that component. Each site will start serial numbers at zero for each component manufactured. Label-making software shall provide a unique site identifier as a prefix to the label serial number.
- Operator Information field identifies the operator who produced the master file. Such identification could be the operator's name and/or badge number or another designation that uniquely identifies the person who performed the mastering operation. The field is a counted ASCII string that starts at byte 254. The string may not exceed 29 bytes in length.
- Mastering Location field identifies where the master file was produced. The field is a counted ASCII string that starts at byte 284. The string may not exceed 29 bytes in length.
- Part Number field contains the part number for the master file. The field is a 17-byte counted ASCII string that starts at byte 314. The string must be in the format: aa-aaaaa-aa.aaaa

  where:

  a represents an alphanumeric character. The string must be exactly 16 bytes in length. The 17th byte of the counted string field must therefore be the null character.
- Data Area Checksum is a word long binary field that starts at byte 332 and is the binary summation of the data in all the blocks of the master file except the header block.
- Header block checksum is a word long binary field that starts at byte 510 and is the binary summation of the preceding 255 words.

The following fields in the header block contain media-specific information. The magtape and disk media master information overlay each other since only one can be in use at a time. The set of fields used in a particular media master file is determined by the value in byte 0, the Master File Format.

The specific information not pertaining to the media builds down from the beginning of the header block. The media-specific information is built up towards the beginning of the header block starting at the word at offset 508. The unused space between the two areas can be allocated to either area in future revisions of the standard provided there is no overlap.

The following fields in the header block are used for magtape media masters.

- Magtape Flags is a word long field at offset 502 that contains a bit mask of logical values specific to magtape media masters. There are no flags defined at this time; the entire word is reserved.
- Tape Type is a byte long binary field at offset 504 that contains a value that is an index into a table of magtape media types. The defined values are:

Value	Tape Type
1	½" 7-track
2	½" 9-track
3	TK50

- Tape Density is a byte long binary field at offset 505 that contains a value that is an index into a table of magtape recording densities. The defined values are:

Value	Tape Density
1	density is N/A (such as for a TK50)
2	800 bpi
3	1600 bpi
4	6250 bpi

The design of this field precludes multiple density tapes. Such tapes are currently illegal for purposes of software distribution.

- Tape Length is a byte long binary field at offset 506 that contains a value that is an index into a table of magtape media lengths. The defined values are:

Value	Tape Length
1	no length choice (such as for a TK50)
2	300 foot
3	600 foot
4	1200 foot
5	2400 foot

- Largest Record Size is a word long binary field at offset 508 that contains the number of bytes of the largest record on the tape. This information can be used in Copy/Verify programs to determine the required buffer size.

The following fields in the header block are used for disk media masters.

- Disk Flags is a word long field at offset 498 that contains a bit mask of logical values that are specific to disk media masters. There are no flags defined at this time. The entire word is reserved.
- Disk Type is a byte long binary field at offset 500 that contains a value that is an index into a table of disk media types. The defined values are:

Value	Disk Type
1	RK05
2	RK06
3	RK07
4	RL01
5	RL02
6	RX01
7	RX02
8	RX24
9	RX31
10	RX33
11	RX50
12	RA60
13	RC25
14	TU58

- Disk Size is a longword binary field starting at offset 502 that contains the size of the disk in logical blocks.
- Disk Geometry is a 2-word binary field starting at offset 506. The first field starting at offset 506 contains the number of sectors per track. The second field starting at offset 508 contains the number of tracks per cylinder. This information can be used to determine the optimum transfer size in Copy/Verify software.

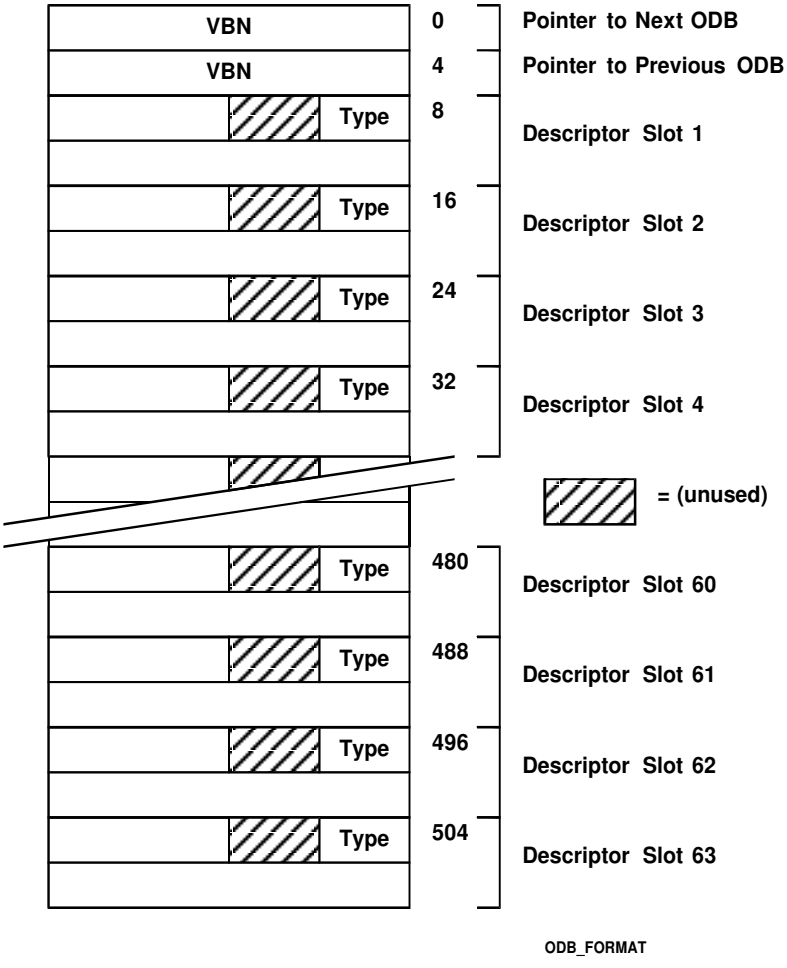
APPENDIX B OBJECT DESCRIPTOR BLOCK

This appendix describes the format of the object descriptor block. Those fields that are common to both magtape and disk are listed and described in subhead B1. Those fields that are unique to a particular media type are listed and described in subheads B2 and B3.

B.1 Format of the Object Descriptor Block

Figure 4 shows the format of the object descriptor block (ODB).

Figure 4: Object Descriptor Block Format



The ODB shall contain the following fields.

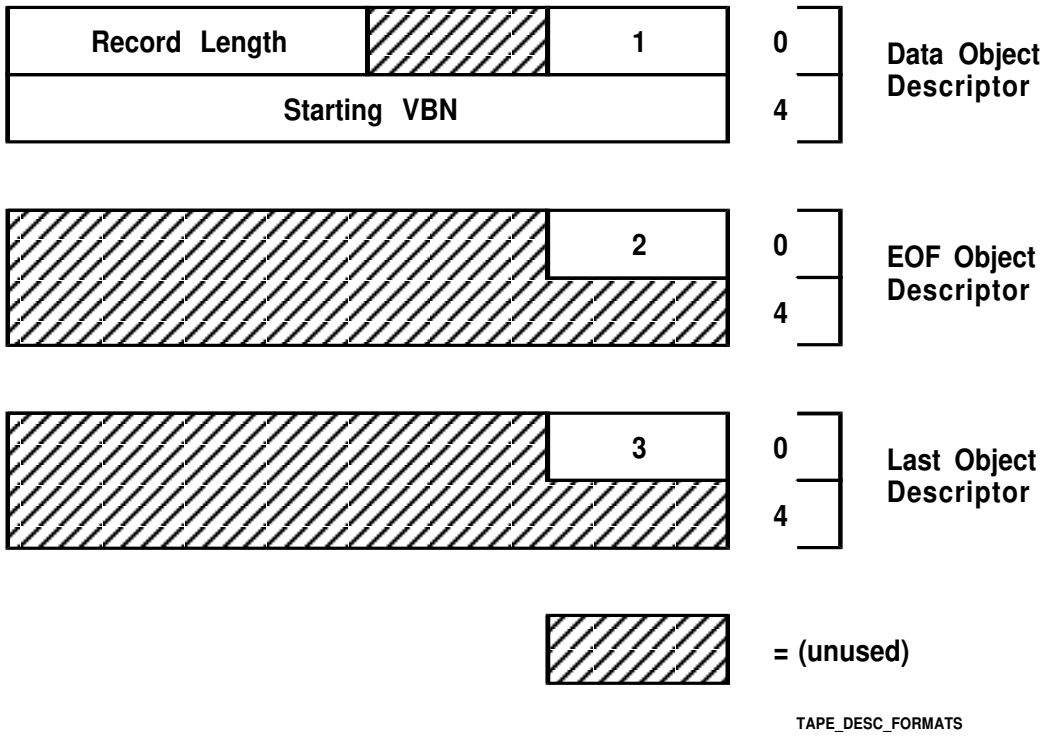
- Next ODB Pointer is a longword binary field starting at offset 0 that contains the virtual block number of the next ODB in the chain. The field contains the value 0 in the case of the last ODB in the chain.
- Previous ODB Pointer is a longword binary field starting at offset 4 that contains the virtual block number of the previous ODB in the chain. The field contains the value 0 in the case of the first ODB in the chain.
- Descriptor Slots comprise the remainder of the ODB, which is divided into 63 descriptor slots. Each slot is two longwords in length and contains descriptors for either tape or disk, depending on the media type of the master file. For tapes, the descriptors record data records and End-Of-File (EOF) marks. For disks, the descriptors record data segments and compression segments. The remaining descriptor type for both disks and tapes denotes logical end of volume.

  When creating a media master file, the mastering program uses descriptor slots in order within an ODB until none remain. The Next ODB Pointer field is filled, and the ODB is written out to the file. A new ODB is initialized in memory, and the descriptor slots within the new ODB are then used.

B.2 Format of the Tape Object Descriptors

Figure 5 shows the formats of the three defined object descriptors used in magtape media master files. All object descriptors are 2 longwords in length. In all cases the byte at offset 0 in each descriptor contains the binary descriptor type field.

Figure 5: Tape Object Descriptor Formats



Type 1 is the Data Object Descriptor. The structure of the descriptor consists of:

- Descriptor Type is a byte-long binary field starting at offset 0 that contains the value 1.
- The byte starting at offset 1 is unused today and reserved for possible future use.
- Record Length is a word-long binary field that contains the length of the data record in bytes. It starts at offset 2 within the descriptor.
- Starting VBN is a longword binary field starting at offset 4 that contains the starting virtual block number in the master file of the data record.

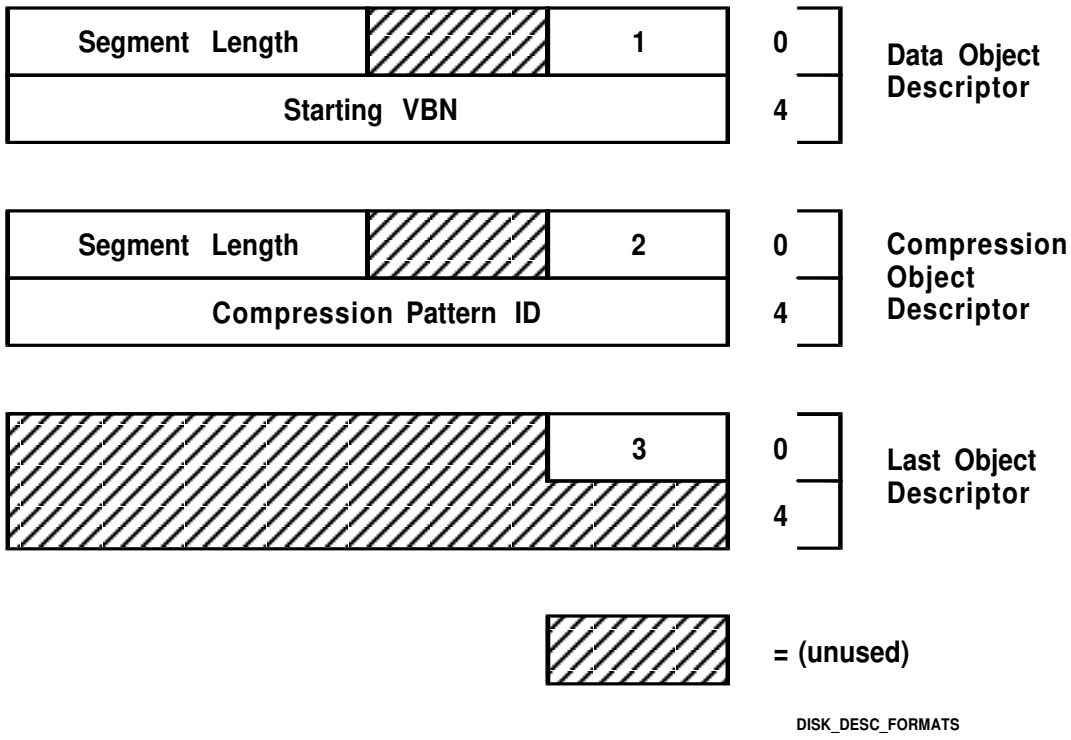
  A data record always starts at the beginning of a block and continues for as many blocks as necessary to hold the record. Any space remaining at the end of the last block is unused. Even though there is some wasted space with this technique, the payback is higher performance since de-blocking of records is not required during the Copy/Verify process.

Descriptor types 2 (EOF Object Descriptor) and 3 (Last Object Descriptor) use only the Descriptor Type field. The remainder of the data structure is unused. Making all descriptors the same length greatly simplifies ODB management at the expense of a little wasted space. This waste is a reasonable tradeoff, however, because there are relatively few EOF marks on a typical magtape and the End-Of-Volume Descriptor is used only once in the file.

B.3 Format of the Disk Object Descriptors

Figure 6 shows the formats of the three defined object descriptors used in disk media master files. All object descriptors are two longwords in length. In all cases the byte at offset 0 in each descriptor contains the binary descriptor type field.

Figure 6: Disk Object Descriptor Formats



Type 1 is the Data Object Descriptor. The structure of the descriptor shall consist of the following.

- Descriptor Type is a one-byte long binary field starting at offset 0 that contains the value 1.
- The byte starting at offset 1 is unused today and reserved for possible future use.
- Segment Length is a word long binary field that contains the length of the data segment in bytes. It starts at offset 2 within the descriptor. Since the segment describes the number of disk blocks to be written, this value should always be a multiple of 512, which is the length of a logical block for all Digital disks.
- Starting VBN is a longword binary field starting at offset 4 that contains the starting virtual block number of the data segment in the master file. The disk Copy/Verify routine shall keep track of the logical block number of the data segment on the output disk by maintaining a running total of the number of blocks written by each data or compression segment.

A segment always starts at the beginning of a block and continues for as many blocks as necessary to hold the data. Therefore, disk media masters, unlike magtape media masters, contain no unused space at the end of the last block.

Type 2 is the Compression Object Descriptor. The structure of the descriptor shall consist of the following.

- Descriptor Type is a byte-long binary field starting at offset 0 that contains the value 1.
- The byte starting at offset 1 is unused today and reserved for possible future use.
- Segment Length is a word-long binary field that contains the length of the data segment in bytes. It starts at offset 2 in the descriptor. Since the segment describes the number of disk blocks to be written, this value should always be a multiple of 512, which is the length of a logical block for all Digital disks.
- Compression Pattern ID is a longword binary field starting at offset 4 that contains the compression pattern identifier. This longword pattern is replicated throughout the segment.

  For example, if the pattern ID is the binary value 0, then the Copy/Verify routine will prepare a zero-filled buffer of the specified length. This buffer is then either copied to the disk at the appropriate starting logical block or compared with the data read from the disk (depending on the mode of operation of the Copy/Verify program).

  The compression pattern ID can be any value and shall be selected by the mastering program. The Copy/Verify program must use the longword found in this field to dynamically build a buffer of the appropriate length and populate it with the replicated longword value. However, certain longword values are likely to appear in many master files and Copy/Verify programs can certainly anticipate these and prepare for them beforehand by pre-allocating and populating buffers with the pattern(s). The pattern values (shown in hexidecimal radix) that are most likely to be seen are:

  - %X'00000000' — (all zeros)
  - %X'FFFFFFFF' — (all ones)
  - %X'DB6DB6DB' — MFM worst-case bit reversal pattern

Descriptor type 3, which is the Last Object Descriptor, uses only the descriptor type field. The remainder of the data structure is unused.

APPENDIX C REFERENCED DOCUMENTS

C.1 EL-Class Digital Documents

EL-Class Number	Document Title
EL-00073-00	*DEC STD 073-0 Manufacturing and Packaging for Publications*

Use VTX SMC to order copies of EL-class documents from Standards and Methods Control. Send distribution questions to JOKUR::SMC or call DTN: 234-4423.

```
+-------------------------------------------------------------------+
|                        READER COMMENTS                            |
|                                                                   |
| Your  comments  and suggestions  will help  Standards and Methods |
| Control improve their services and documents.                     |
+-------------------------------------------------------------------+
```

Did you request this document? _____ If so, did it arrive within a satisfactory period of time? _____ Please comment.

What are your impressions of this document? Consider format, organization, completeness, readability, and illustrations.

————————**FOLD ON THIS LINE**———————————————————— -

Did you find technical or clerical errors in this document? If so, please specify the page number(s) and the error(s).

Are the instructions for the update package clear? _____
Was an index available? _____ If not, is one needed? _____
Do you have other suggestions for improving this document?

The following information is optional:

Name _____ Mailstop _____

Department _____ Node _____

Send your comments to JOKUR::PROJECTS, or fold, staple, and send this page through interoffice mail to:

```
+------------------------------+
|       READERS' COMMENTS      |
| STANDARDS AND METHODS CONTROL |
|           NRO4/D4            |
+------------------------------+
```